

IN THE CLAIMS

Please amend the following claims:

- Sub  
A137
1. (Amended) A method for aligning an [macro] instruction stream comprising:
  - 2 rotating data bytes of the [macro] instruction stream; and
  - 3 shifting the data bytes to a [the] start of the [a macro] instruction based upon a
  - 4 length of an immediately prior [macro] instruction.

- Sub  
B8  
A2
- 1 3. (Amended) The method of claim 1 further comprising:
  - 2 receiving a length of an immediately prior [macro] instruction from a length
  - 3 decode logic unit.
  - 1 4. (Amended) The method of claim 1 further comprising:
  - 2 storing [macro] instruction stream cache lines in alignment buffers prior to
  - 3 rotating the [macro] instruction stream.
  - 1 5. (Amended) The method of claim 1 wherein said shifting shifts to an exact start
  - 2 of the [macro] instruction.

- Sub  
A137
- 1 7. (Amended) Logic for aligning an [macro] instruction stream comprising:
  - 2 a rotator logic unit for rotating data bytes of the [macro] instruction
  - 3 stream;
  - 4 a shifter logic unit for shifting the data bytes to a [the] start of the [a
  - 5 macro] instruction based upon a length of an immediately prior [macro]
  - 6 instruction.

Sub  
Bio

1 9. (Amended) The logic of claim 7 further comprising:  
2 a length vector for providing the length of an immediately prior [macro]  
3 instruction.

1 10. (Amended) The logic of claim 7 further comprising:  
2 alignment buffers for storing [macro] instruction stream cache lines for  
3 use by the rotator logic unit.

1 11. (Amended) A processor to align an [macro] instruction stream comprising:  
2 a rotator logic unit for rotating data bytes of the [macro] instruction  
3 stream;  
4 a shifter logic unit for shifting the data bytes to a [the] start of the [a  
5 macro] instruction based upon a length of an immediately prior [macro]  
6 instruction.

1 12. (Amended) The processor of claim, 11 further comprising:  
2 a length vector for providing the length of an immediately prior [macro]  
3 instruction.

1 13. (Amended) A system for aligning an [macro] instruction stream comprising:  
2 means for rotating data bytes of the [macro] instruction stream; and  
3 means for shifting the data bytes to a [the] start of the [a macro]  
4 instruction based upon a length of an immediately prior [macro] instruction.

Please add the following claims:

- Sub 11  
AS
- 1 14. The method of claim 1 further comprising:  
2 determining a length of a current instruction.
- 1 15. The method of claim 14 wherein the length of the current instruction is based  
2 upon a length of an opcode and a length of immediate data.
- 1 16. The method of claim 14 further comprising:  
2 determining if an opcode extension byte is required to determine the  
3 length of the current instruction.
- 1 17. The method of claim 16 further comprising:  
2 determining a memory address displacement length.
- 1 18. The method of claim 17 further comprising:  
2 determining an anticipatory length of the memory displacement for a one-  
3 byte opcode; and  
4 determining an anticipatory length of the memory displacement for a two-  
5 byte opcode.
- 1 19. The method of claim 18 further comprising multiplexing the anticipatory  
2 length for the one-byte opcode and the anticipatory length for the two-byte  
3 opcode to determine the length of the memory displacement.